



# »Лекционен курс »ООП1 (Java)

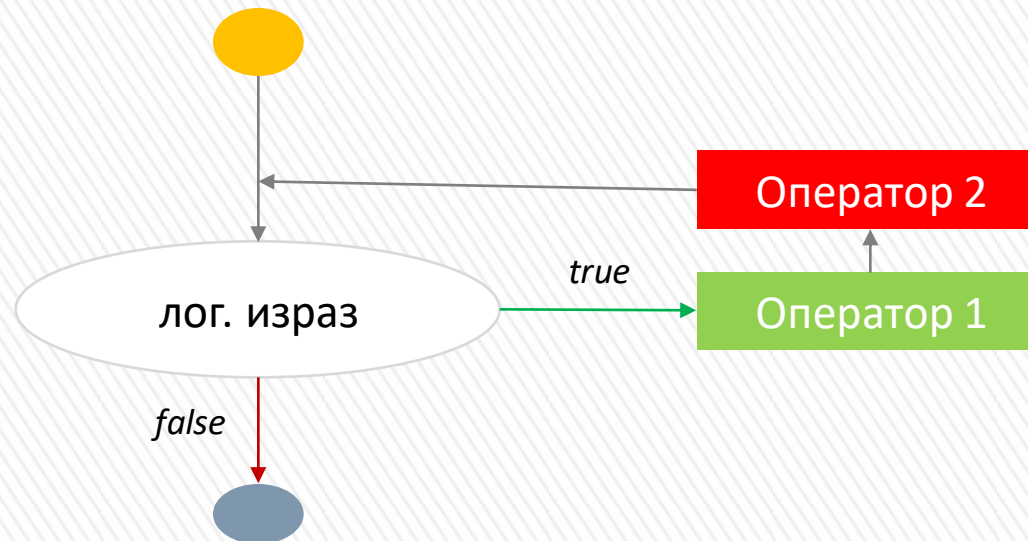


## Итерации



# while оператор: циклична структура

```
while (логически израз) {  
    оператор 1;  
    оператор 2;  
}
```



# Синтаксис

EBNF:

**while** ( условие ) оператор

Докато 'условие' е изпълнено, повтори  
'оператор'

**Съществен специален случай:**  
операторът не се обработва

# Пример



Какво прави програмата?

Брои

```
import java.util.Scanner;
public class WhileDemo {
    public static void main(String[] args) {
        int count, number;
        System.out.println("Въведи число");
        Scanner keyboard = new Scanner(System.in);
        number = keyboard.nextInt();
        count = 1;
        while (count <= number) {
            System.out.print(count + ", ");
            count++;
        }
        System.out.println();
        System.out.println("Край.");
    }
}
```

# Пример



Какъв резултат?

```
import java.util.Scanner;
public class WhileDemo {
    public static void main(String[] args) {
        int count, number;
        System.out.println("Въведи  
число");
        Scanner keyboard = new Scanner(System.in);
        number = keyboard.nextInt();
        count = 1;
        while (count <= number) {
            System.out.print(count + " ");
            count++;
        }
        System.out.println();
        System.out.println("Край.");
    }
}
```

Въведи число

2

1, 2,

Край.

Process finished with exit code 0

Въведи число

3

1, 2, 3,

Край.

Process finished with exit code 0

Въведи число

0

Край.

Process finished with exit code 0

# do-while-оператор

EBNF:

```
do оператор while ( условие ) ;
```

**Докато 'условие' е изпълнено, повтори 'оператор', при което условието се тества след изпълнение на оператора**

**Приложение:**

Операторът се обработва поне веднъж



# Пример



Какво прави програмата?

Брой

```
import java.util.Scanner;
public class DoWhileDemo {
    public static void main(String[] args) {
        int count, number;
        System.out.println("Въведи число:");
        Scanner keyboard = new Scanner(System.in);
        number = keyboard.nextInt();
        count = 1;
        do {
            System.out.print(count + ", ");
            count++;
        } while (count <= number);
        System.out.println();
        System.out.println("Край.");
    }
}
```

# Пример



Какъв резултат?

```
import java.util.Scanner;
public class DoWhileDemo {
    public static void main(String[] args) {
        int count, number;
        System.out.println("Въведи число:");
        Scanner keyboard = new Scanner(System.in);
        number = keyboard.nextInt();
        count = 1;
        do {
            System.out.print(count);
            count++;
        } while (count <= number);
        System.out.println();
        System.out.println("Край.");
    }
}
```

Въведи число:

2

1, 2,

Край.

Process finished with exit code 0

Въведи число:

3

1, 2, 3,

Край.

Process finished with exit code 0

Въведи число:

0

1,

Край.

Process finished with exit code 0



# do-while-оператор



Последователен ли е синтаксисът?

EBNF:

`do оператор while ( условие ) ;`

Докато 'условие' е изпълнено, повтори 'оператор', при което условието се тества след изпълнение на оператора

Приложение:

Операторът се обработва поне веднъж

# do-while-оператор: представлен чрез while-оператор



Възможно ли е?

```
do  
    оператор  
while ( условие ) ;
```

Еквивалентен на:

```
оператор  
while ( условие )  
    оператор
```

# Роля на ';' : Разделяне или край на оператори ?

Pascal, Modula-2, Ada, ... :  
**Разделяне на оператори**

```
x := y ;  
if x > y then  
  begin x := y ; y := 0 end ;  
d := 100
```

# Ролята на ';'

C, C++, Java:

- Край на оператори  
(синтактически – част от операторите)
- но: с много изключения  
(без ';' : while, if, ...)

→ “нечиста” езикова дефиниция (неединен принцип: причина за грешки)

```
x = y;
```

```
if (x > y) {
```

```
    x = y; y = 0;
```

```
}
```

```
d = 100;
```

# Пример: функцията “!”

$$n! = 1 * 2 * \dots * n$$

```
int n, x = 1, fac = 1;
... // read n
while (x <= n) {
    fac = fac * x;
    x = x + 1;
}
```

напр. за  $n = 4$ :

- $x$  получава стойностите: 1, 2, 3, 4, 5  
(при  $x = 5$  не важи повече:  $x \leq n$ )
- При това  $fac$  получава стойностите: 1, 1, 2, 6, 24

# 1 ½-цикъл-проблем

**Break-оператор:** ИЗХОД ОТ ЦИКЪЛ В НЕГОВОТО ТЯЛО

```
while (true) {  
    ... четене ...  
    if (Keyboard.eof())  
        break;  
    ... обработка входа ...  
}
```



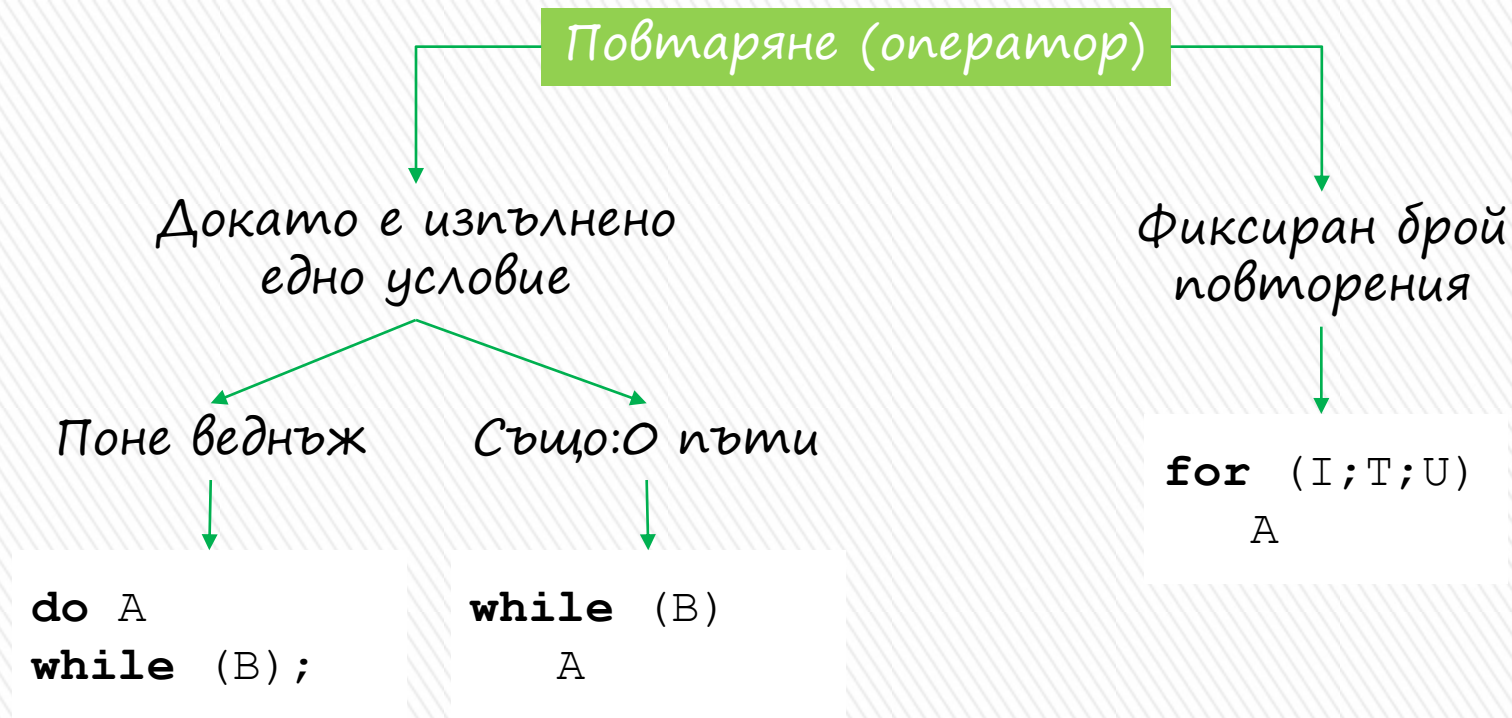
# Break-оператор:

## " 1 ½ - цикъл-проблем"

*Също без break:*

```
... четене ...  
while (! Keyboard.eof())  
    ... Обработка входа ...  
    ... четене ...  
}
```

# Избор на повтаряеми оператори



# For-цикъл

EBNF:

**for** (инициализиране ; Test ; Update) оператор

Изрази (обикновено присвоявания)



Еквивалентно на:

```
инициализиране ;  
while ( Test ) {  
    оператор ;  
    Update ;  
}
```

# Пример



Какво прави програмата?

Брои преди излитане

```
public class ForDemo {  
    public static void main(String[] args) {  
        int countDown;  
        for (countDown = 3; countDown >= 0; countDown--) {  
            System.out.println(countDown);  
            System.out.println("... и броя.");  
        }  
        System.out.println("Излитам!");  
    }  
}
```

# Пример



Какъв резултат?

```
public class ForDemo {  
    public static void main(String[] args) {  
        int countDown;  
        for (countDown = 3; countDown >= 0; countDown--) {  
            System.out.println(countDown);  
            System.out.println("... и броя.");  
        }  
        System.out.println("Излитам!");  
    }  
}
```

```
3  
... и броя.  
2  
... и броя.  
1  
... и броя.  
0  
... и броя.  
Излитам!
```

```
Process finished with exit code 0
```



# Пример: ! с 'for'

```
int n, x, fac = 1;  
... // read n  
for (x = 1; x <= n ; x = x + 1)  
    fac = fac * x;
```

Кратка форма:  $x++$

Аналог на:  $x--$  като  $x = x - 1$

Приложение: броят на повторенията е познат  
(тук:  $n$ ) !



# Локални декларации



Възможно ли е?

```
int n, x, fac = 1;  
... // read n  
for (x = 1; x <= n ; x = x + 1)  
    fac = fac * x;
```

```
int n, fac = 1;  
... // read n  
for (int x = 1; x <= n ; x = x + 1)  
    fac = fac * x;
```

В Java можем да разпръснем  
декларирането на  
променливите в рамките на  
целия блок, като ги  
дефинираме в момента, в  
който ни потрябват

# Тест: функцията “!”



Какъв резултат?

```
public static int faculty (int n) {  
    int x, fac=1;  
  
    for (x=1; x<=n; x++)  
        fac = fac * x;  
  
    return fac;  
}  
  
public static void main (String[] args) {  
  
    int x, y;  
  
    x = faculty(3);  
    System.out.println("3! = " + x);  
    y = faculty(5);  
    System.out.println("5! = " + y);  
}  
}
```

**3! = 6**  
**5! = 120**

# Пример за цикъл

Таблица: Трансформиране Celsius → Fahrenheit

Grad C	Grad F
-10.0	14.0
-9.0	15.8
-8.0	17.6
...	
10.0	50.0

Примерни програми с *while* и *for* оператор

# C while-оператор

```
class TemperatureTable {
    // Table with C/F temperatures
    public static void main (String[] args) {
        final double
            LOW_TEMP = -10.0,
            HIGH_TEMP = 10.0;

        double
            cent, // degree Celsius
            fahr; // degree Fahrenheit

        System.out.println("\tGrad C\t\t\tGrad F");
        cent = LOW_TEMP;
        while (cent <= HIGH_TEMP) {
            fahr = (9.0/5.0) * cent + 32.0; // C -> F
            System.out.println("\t" + cent + "\t\t" + fahr);
            cent = cent + 1.0;
        }
    }
}
```

# C for-оператор

```
class TemperatureTable {
    // Table with C/F temperatures
    public static void main (String[] args) {
        final double
            LOW_TEMP = -10.0,
            HIGH_TEMP = 10.0;

        double
            cent, // Grad Celsius
            fahr; // Grad Fahrenheit

        System.out.println("\tGrad C\t\tGrad F");
        for(cent = LOW_TEMP; cent <=HIGH_TEMP; cent++) {
            fahr = (9.0/5.0) * cent + 32.0; // C -> F
            System.out.println("\t" + cent + "\t\t" + fahr);
        }
    }
}
```

# Оператор „,”

```
public class CommaOperator {  
    public static void main(String[] args) {  
        for( int i=1, j=i+10; i<5; i++, j=i*2) {  
            System.out.println("i= " + i + " j= " + j);  
        }  
    }  
}
```

*Оператор „,” (последователност): има само едно приложение в Java – в управляващия израз на **for***

*В управляващия блок на цикъла:*

- Допуска се дефиниране на повече от една променлива, но те трябва да бъдат от един и същ тип
- Променливата се дефинира там, където се използва
- Областта на видимост – израз, управляващ **for**



# Оператор „,”



Какъв резултат?

```
public class CommaOperator {  
    public static void main(String[] args) {  
        for( int i=1, j=i+10; i<5; i++, j=i*2) {  
            System.out.println("i= " + i + " j= " + j);  
        }  
    }  
}
```

```
i= 1 j= 11  
i= 2 j= 4  
i= 3 j= 6  
i= 4 j= 8
```

# Break & continue

» В тялото на всяка една от управляващите конструкции можем да управляваме хода на цикъла посредством операторите `break` и `continue`

- > `break` – излиза от цикъла без да се изпълняват останалите конструкции в него
- > `continue` – спира изпълнението на текущата итерация и се връща в началото на цикъла, за да започне изпълнение на следващата итерация

# Пример



Какъв резултат?

```
public class BreeakAndContinue {
    public static void main(String[] args) {
        for(int i = 0; i < 100; i++) {
            if(i == 74) break; // Излиза от цикъла for
            if(i % 9 != 0) continue; // Следваща итерация
            System.out.println(i);
        }
        int i = 0;
        // "Безкраен цикъл":
        while(true) {
            i++;
            int j = i * 27;
            if(j == 1269) break; // Излиза от цикъла
            if(i % 10 != 0) continue; // Връщане в
            // началото на цикъла
            System.out.println(i);
        }
    }
}
```

0  
9  
18  
27  
36  
45  
54  
63  
72

10  
20  
30  
40

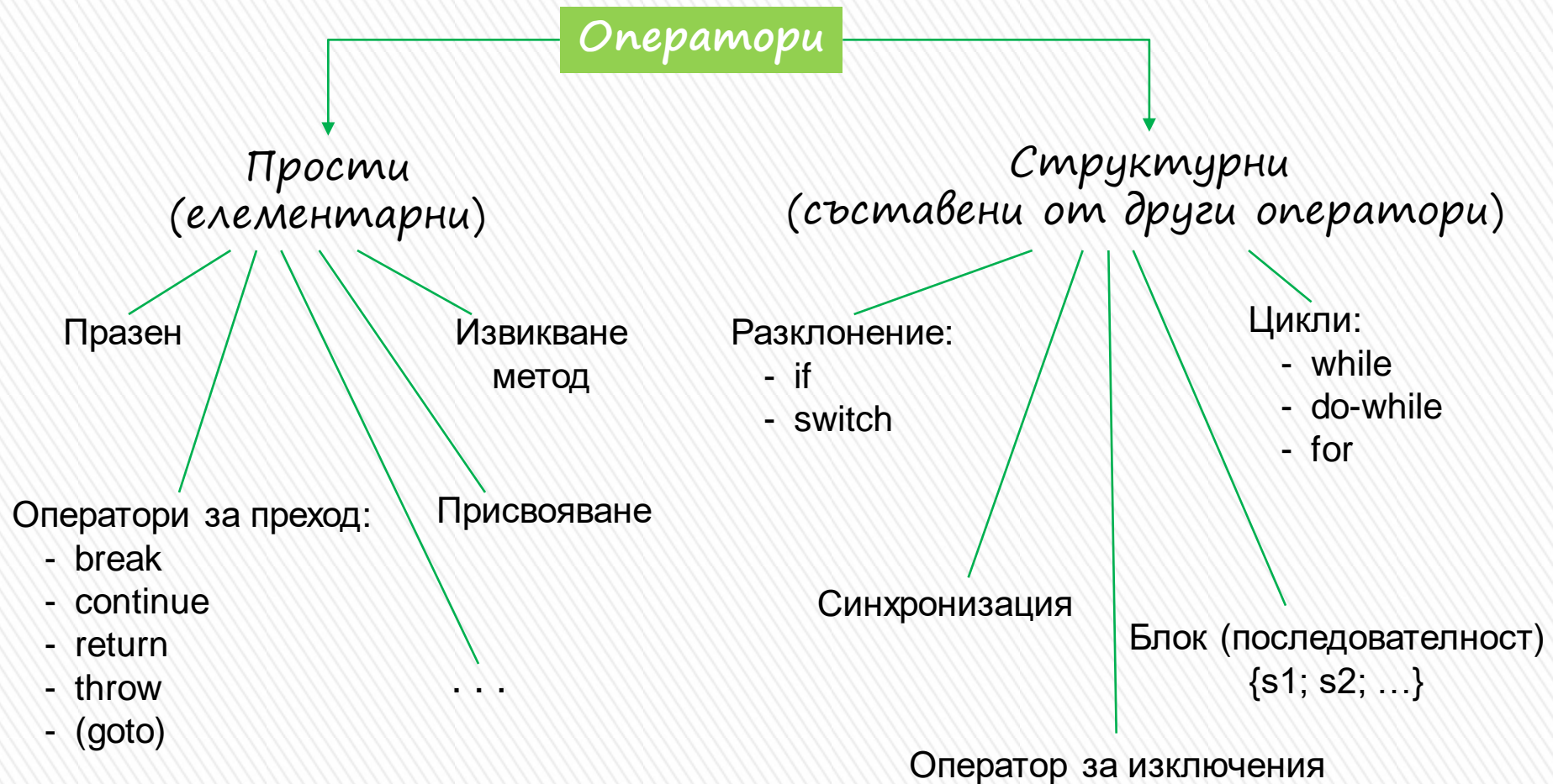


# Безкрайни цикли

`while(true)`

`for(;;)`

# Структурно програмиране: операторите



```

import java.util.Scanner;
public class ExamAverager {
    public static void main(String[] args) {
        System.out.println("Тази програма изчислява средния успех");
        System.out.println("при въведен списък с оценки.");
        double sum;
        int numberOfStudents;
        double next;
        String answer;
        Scanner keyboard = new Scanner(System.in);
        do {
            System.out.println();
            System.out.println("Въведи всички оценки.");
            System.out.println("Въведи отрицателно число за край.");
            sum = 0;
            numberOfStudents = 0;
            next = keyboard.nextDouble();
            while (next >= 0) {
                sum = sum + next;
                numberOfStudents++;
                next = keyboard.nextDouble();
            }
            if (numberOfStudents > 0)
                System.out.println("Средният успех е " + (sum/numberOfStudents));
            else
                System.out.println("Не са въведени оценки.");
            System.out.println("Желаеш ли среден успех за друг студент?");
            System.out.println("Въведи , да ` или , не `.");
            answer = keyboard.next();
        } while (answer.equalsIgnoreCase("да"));
    }
}

```



Какво прави програмата?

Изчислява  
среден успех





```

import java.util.Scanner;
public class ExamAverager {
    public static void main(String[] args) {
        System.out.println("Тази програма изчислява средния успех");
        System.out.println("при въведен списък с оценки.");
        double sum;
        int numberOfStudents;
        double next;
        String answer;
        Scanner keyboard = new Scanner(System.in);
        do {
            System.out.println();
            System.out.println("Въведи всички оценки.");
            System.out.println("Въведи отрицателно число за край.");
            sum = 0;
            numberOfStudents = 0;
            next = keyboard.nextDouble();
            while (next >= 0) {
                sum = sum + next;
                numberOfStudents++;
                next = keyboard.nextDouble();
            }
            if (numberOfStudents > 0)
                System.out.println("Средният успех е " + sum / numberOfStudents);
            else
                System.out.println("Не са въведени оценки.");
            System.out.println("Желаеш ли среден успех за друг студент?");
            System.out.println("Въведи ,да` или ,не`.");
            answer = keyboard.next();
        } while (answer.equalsIgnoreCase("да"));
    }
}

```



Какъв резултат?

```

Тази програма изчислява средния успех
при въведен списък с оценки.

Въведи всички оценки.
Въведи отрицателно число за край.

3
4
6
6
-1
Средният успех е 4.75
Желаеш ли среден успех за друг студент?
Въведи ,да` или ,не`.
не

Process finished with exit code 0

```



*Благодаря за вниманието!*